Solution 1> Given : x and a highly-unlikely-and chosen δ

It is difficult to find z such that $h(\delta|z) = x$ (but it should exist)

Now in the given case:

$G(z) = H(z) || z_{last}$

To show : G is puzzle friendly and not hiding.

Given: $G(z) = H(z) || z_{last}$: z_{last} = last bit of z

Puzzle Friendlyness: we say there is x and a highly-unlikely and randomly Chosen δ.

It is difficult to find z such that $h(\delta|z) = x$ (but it should exist)

Now in the given case.

$G(z) = H(z) || z_{last}$

But it's already given H(z) is puzzle friendly, in for a given x and δ it is difficult to find z in $H(\delta|z) = x$, so if we can't find z from H then it is difficult to find z_{last} as well and hence from G(z) we can't find z easily for the given x and δ it is also Puzzle friendly.

Hiding: A hash function is hiding if secret value s is chosen from a probability distribution that has high entropy, then given $H(s||x)$ it's infeasible to find x.

Given: $H(s|x)$

Secret: x and a Highly - unlikely and randomly chosen s.

How to find y such that $H(y) = H(s|x)$

The hiding property should work for all plaintext spaces, even if your plaintext space is $\{0,1\}$. G is not hiding for that plaintext space. That means if $z=0$ or 1 then for

$$G(z) = H(z) || z_{last}$$

$z_{last} = z$, so for given $G(z)$ last bit is z and hiding property does not hold for one bit numbers.

* SHA-256 (Secure Hashing Algorithm)

It is one of the cryptographic hash function which has digest length of 256 bits. It's a keyless hash function. It was developed by National Institutes of Standards & Technology.

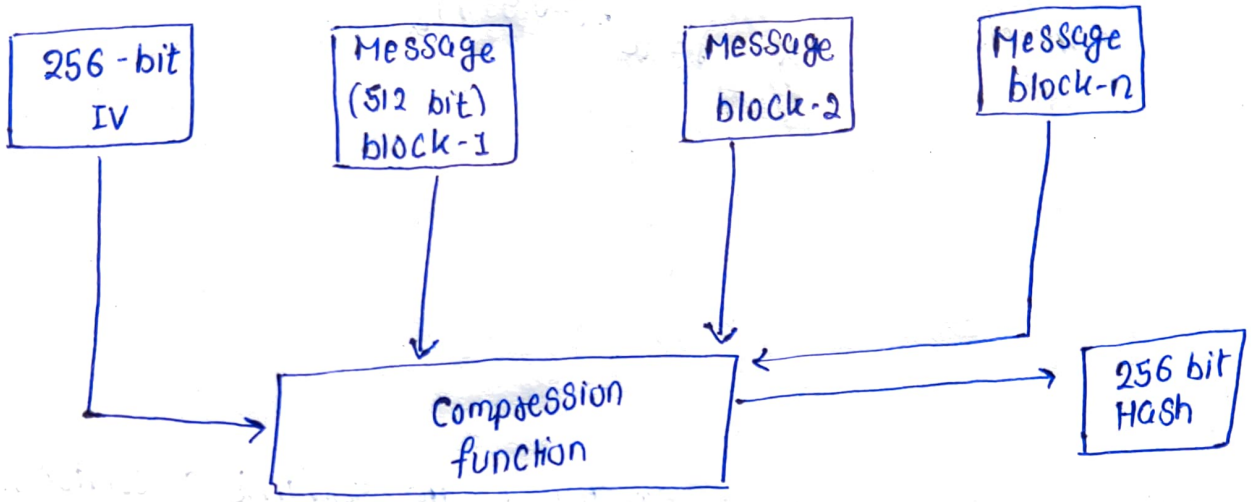
These are Five ~~requirements~~ properties for SHA256:-

1) It is one way - cannot restore data from hash value.
2) It is deterministic
3) Fast Computation
4) The Avalanche effect
5) Must withstand collisions.

Name: Lokhan Kumawat
Roll No: 1906055
Branch: CSE-1

Course Code: Blockchain Technology
Course: CS6475.

08/03/2022 1.



Name: Lakhan Kumawat
Roll No: 1906055
Branch: CSE-1

Course Code: Blockchain Technology
Course: CS6475.

08/03/2022

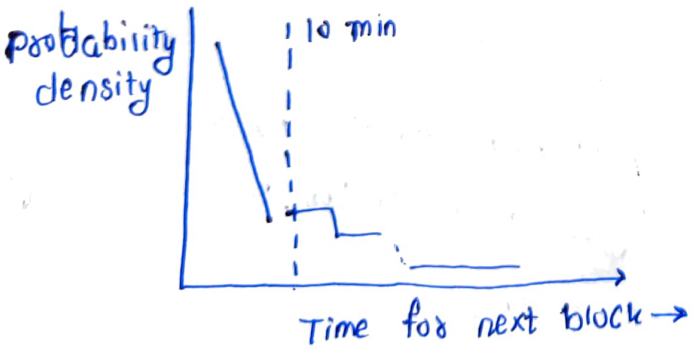
Solution 2:

a) No. the transactions, that users create all require a digital signature. Creating a valid digital signature requires the private key for the public key Because of that public key is fixed, the specific private key that the user has is required in order to modify the transaction.

An ISP doesn't have that user's private keys. so they will be unable to produce a coflicting transaction. They cannot produce a valid signature, so a double spend cannot be made.

The only thing ISP can do is censor a user's transactions.

If a block is found now than the next block will be found soon and there is a small probability that it will take a long time to find the next block.



Mean time to find a block = 10 minutes / fraction of hash power.

∴ probability to find the next block in next 10 min = more than 50 %.

3.

Solution 3> probability of n blocks is found in the time that we expected λ blocks to be found is $P(x/\lambda) = \frac{e^{-\lambda}\lambda^x}{x!}$

Given question, we need 6 blocks to be bound with a confidence or probability of 0.99.

So $P(x \geq 6/\lambda) = 0.99$

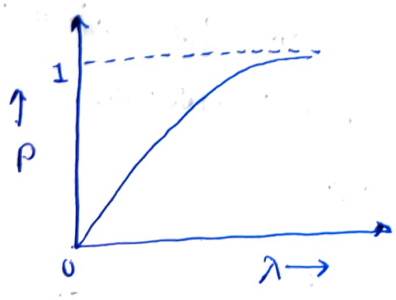
But, $P(x \geq 6/\lambda) = 1 - P(x=5/\lambda) - P(x=4/\lambda) - P(x=3/\lambda) - P(x=2/\lambda) - P(x=1/\lambda) - P(x=0/\lambda)$

$1 - P(x=5/\lambda) - P(x=4/\lambda) - P(x=3/\lambda) - P(x=2/\lambda) - P(x=1/\lambda) - P(x=0/\lambda) = 0.99$

$\sum_{\lambda=0}^{5} P(x=\lambda/\lambda) = 0.01$

$$e^{-\lambda}\left[\frac{\lambda^0}{0!} + \frac{\lambda^1}{1!} + \frac{\lambda^2}{2!} + \frac{\lambda^3}{3!} + \frac{\lambda^4}{4!} + \frac{\lambda^5}{5!}\right] = \frac{1}{100}$$

The probability curve looks like



So by Soving the above equation on the Scientific Calculator, it results to be λ is approximately equal to 13.1085.

So, by solving the above equation on the scientific calculator, it results to be λ is approximately equal to 13.1085.

That means it takes 13.1085 blocks time to find atleast 6 block with 99% confidence.

But in question, it is stated each block take time 10 min to create so, approximately 131 minutes it takes for 99% of the cases atleast 6 blocks will have been found.

Name: Lakhan Kumawat
Roll No: 1906055
Branch: CSE-1

Course Code: Blockchain Technology
Course: CS6475.

08/03/2022

14

That means it takes 13.1085 blocks time to find atleast 6 blocks with 99% Confidence.

But in question it is stated that each block take 10 mins to create so approximately 131 minutes will take for 99% of the cases atleast 6 blocks have been found.

Solution

4) i) Digital Signature attacks:-

These are three types of digital signature attacks :-

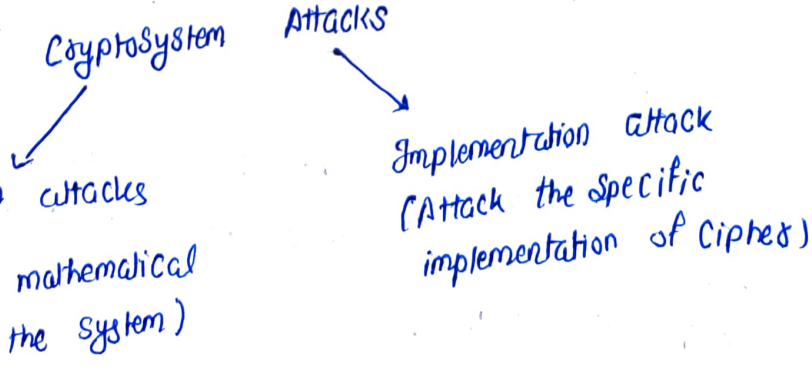
a) Choosen Message attack :- The attacker tricks the genuine user to digitally sign on message that user does not normally intend to sign. Then attacker tries to create a new message that he/she wants a genuine use to sign and use previous signature e.

b) Known meggssage attack :-

The attackers c Some messages that user sends and a key to create a new fault message and forge of user.

c) key only attack

This it is assumed that user name some information public and attacker tries to misuse the public information.



Following attacks fall in one of above two cases.

→ Cipher text only attack:

Attacker knows only the cipher text to be decoded. the attacker will try to find the key or decrypt one or more pieces of cipher text.

→ known plain text attack:-

Attacker has collection of plain text, cipher text trying to find the key or decrypt some other cipher text

→ Chosen cipher text attack:-

· Ability to select any. Cipher text & study the plaintext produced by decryption.

→ Chosen testattack:-

Has ability to dequired in previous two attacks

4) ii)

$p = 809$, $q = 751$, $d = 23$

$n = p \times q = 809 \times 751$

$= 607559$

Public key e: Can be calculated as

$d \times e = 1 \bmod (n)$

$23 \times e = 1 \bmod 606000$

$e = 158087$

Name: Lakhan Kumawat
Roll No: 1906055
Branch: CSE-1

Course Code: Blockchain Technology
Course: CS6475.

08/03/2022 1.

16.

with private key & decrypt with public key.

a) Signing of $M1 = 100$ as:

$$S_1 = M_1^d \bmod n = 100^{23} \bmod 607559$$

$$= 223388$$

Verification of $S1 = 223388$ as

$$M_1 = S_1^e \bmod n = 223388^{258087} \bmod 607559$$

$$= 100$$

b) Signing of $M_2 = 50$ as

$$S_2 = M_2^d \bmod n = 50^{23} \bmod 607559 = 5627$$

verification of $S_2 = 5627$ as

$$M_2 = S_2^e \bmod n = 5627^{15807} \bmod 607559$$

$$= 50.$$

c) $M = M_1 \times M_2 = 5000$

$$S = S_1 \times S_2$$

$$= (223388 \times 5027) \bmod 607559$$

$$S = 572264$$

$$S = M^d \bmod 607559$$

$$S = 5000^{23} \bmod 607559$$

$$S = 572264$$

Name: Lokhan Kumawat
Roll No: 1906055
Branch: CSE-1

Course Code: Blockchain Technology
Course: CS6475.
08/03/2022

File Name: arrays.sol

Solution 5:)

```
//SPDX -License -Identifier : GPL-3.0
Pragma Solidity ^0.8.11;
Contract myContract{

uint arr[];

//add element at the element of array

function addElement (int x) public{
        arr.push(x);
}

// get element at the given index
function getElement (uint idx) public view returns (uint)
{
    if( idx <arr.length())
        return arr[idx];

    return -1;
}

//update element at the given index
function updateElement (int x, uint idx)
Dubic
  {
    arr[idx]=x;
  }
}
```